Ref. 1

(19)日本国特許庁(JP)　(12)公開特許公報（A）　(11)特許出願公開番号

特開平１０－３０１８８０

(43)公開日　平成10年(1998)11月13日

(51)Int. Cl.[6]		識別記号	FI		
G06F	13/00	355	G06F	13/00	355
	12/00	546		12/00	546 K
	15/163			15/16	320 K

審査請求　未請求　請求項の数２０　ＯＬ　（全２２頁）

(21)出願番号　特願平10-91588

(22)出願日　平成10年(1998)4月3日

(31)優先権主張番号　08/827,763

(32)優先日　1997年4月10日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390035493
エイ・ティ・アンド・ティ・コーポレーション
ＡＴ&Ｔ　ＣＯＲＰ.
アメリカ合衆国　10013－2412　ニューヨーク　ニューヨーク　アヴェニュー　オブ　ジ　アメリカズ　32

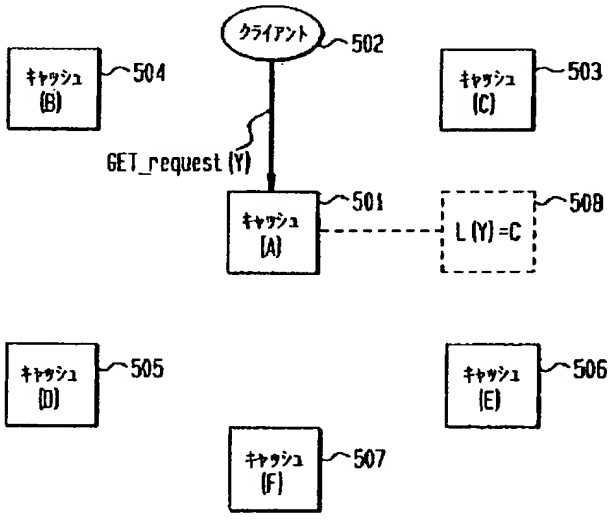
(72)発明者　ジョン　ジェイ　チャン
アメリカ合衆国　カリフォルニア州　サンフランシスコ　21エスティ　アヴェニュー　1630

(74)代理人　弁理士　吉田　研二　（外2名）

最終頁に続く

(54)【発明の名称】スケーラブルネットワークオブジェクトキャッシュ

(57)【要約】
【課題】　スケーラビリティに優れたネットワーク用分散型キャッシュシステムを提供する。
【解決手段】　ネットワーク上にオブジェクトキャッシュ５０３～５０７を配置しておき、そのいくつかにオリジナル情報源からのデータオブジェクトのコピーを記憶させておく。受信キャッシュ５０１にクライアント５０２からあるオブジェクトに対するリクエストがあれば、そのオブジェクトに関するディレクトリリストがどのオブジェクトキャッシュ５０３～５０７に記憶されているかをディレクトリロケータ関数５０８を実行することにより求め、受信キャッシュ５０１はその求められたキャッシュにデータオブジェクトのコピーが何処に記憶されているかを問い合わせる。



1

【特許請求の範囲】

【請求項１】　ネットワークにおける分散型キャッシュシステムであって、

プロセッサとコンピュータにより読み出し可能なメモリとネットワークに接続するポートとを接続するデータバスを含み、ユーザからオブジェクトに対するリクエストを受信し、前記リクエストを入力として用いてポインタを出力として与えるディレクトリロケータ関数を実行する、受信キャッシュと、

プロセッサとコンピュータにより読み出し可能なメモリとネットワークに接続するポートとを接続するデータバスを含み、前記ポインタがネットワークアドレスを示す場合に前記ポインタによってそのネットワークアドレスが示され、前記メモリ中にリクエストされた前記オブジェクトのディレクトリリストを記憶する、ディレクトリキャッシュと、を含み、

前記ディレクトリリストは、リクエストされた前記オブジェクトのネットワークアドレスと、ネットワーク上のオブジェクトキャッシュのネットワークアドレスとを含み、

前記オブジェクトキャッシュは、プロセッサとコンピュータにより読み出し可能なメモリとネットワークに接続するポートとを接続するデータバスを含み、前記メモリ中にリクエストされた前記オブジェクトのコピーを記憶するシステム。

【請求項２】　請求項１に記載のシステムにおいて、前記受信キャッシュは、前記ポインタがネットワークアドレスを示さない場合、前記ユーザによってリクエストされたオブジェクトに対するリクエストを前記オブジェクトのネットワークアドレスに送信するシステム。

【請求項３】　請求項１に記載のシステムにおいて、前記受信キャッシュ、前記ディレクトリキャッシュ、および前記オブジェクトキャッシュは、ネットワーク上の階層レベルの第１レベルで論理上グループ分けされ、

前記システムはさらに、前記階層レベルの第２レベルで論理上グループ分けされる第２の受信キャッシュと、第２のディレクトリキャッシュと、第２のオブジェクトキャッシュとを含み、

前記第１レベルの前記受信キャッシュは、前記ポインタがネットワークアドレスを示さない場合、前記ユーザからリクエストされたオブジェクトに対するリクエストを前記第２レベルの前記第２の受信キャッシュに送信するシステム。

【請求項４】　ネットワーク上に記憶されたデータオブジェクトに対するユーザからのリクエストを受信する、ネットワーク上のスケーラブルな分散型キャッシングシステムを実現する方法であって、

データオブジェクトのコピーをネットワーク上のオブジェクトキャッシュに記憶するステップと、

2

ディレクトリリストをネットワーク上のディレクトリキャッシュに記憶するステップであって、前記ディレクトリリストは、前記オブジェクトのネットワークアドレスと、前記オブジェクトのコピーを記憶していると称されるオブジェクトキャッシュのネットワークアドレスとを含むステップと、

ユーザからオブジェクトに対するリクエストを受信する受信キャッシュ上でロケータ関数を実行するステップであって、前記リクエストは前記オブジェクトのネットワークアドレスを含み、前記ロケータ関数は前記ユーザリクエストを入力として使用してポインタを出力として与えるステップと、を含む方法。

【請求項５】　請求項４に記載の方法において、前記ポインタはネットワークアドレスを示さない方法。

【請求項６】　請求項５に記載の方法であって、

前記受信キャッシュから前記オブジェクトの前記ネットワークアドレスにおけるサーバに、前記オブジェクトのコピーを要求するメッセージを送信するステップと、

前記オブジェクトの前記ネットワークアドレスにおける前記サーバから前記受信キャッシュに、前記オブジェクトのコピーを送信するステップと、をさらに含む方法。

【請求項７】　請求項６に記載の方法であって、

前記受信キャッシュに前記オブジェクトのコピーを記憶するステップと、

前記受信キャッシュから前記ディレクトリキャッシュに、前記オブジェクトのディレクトリキャッシュ上の前記オブジェクトキャッシュのアドレスを削除し、前記受信キャッシュのアドレスを前記ディレクトリリストに加えることを要求するメッセージを送信するステップと、

前記オブジェクトの前記ディレクトリキャッシュ上の前記オブジェクトキャッシュのアドレスを削除し、前記受信キャッシュのアドレスを前記ディレクトリリストに加えるステップと、をさらに含む方法。

【請求項８】　請求項５に記載の方法において、前記受信キャッシュは、ネットワーク上のキャッシュの論理レベルの階層中の第１の論理レベルにあり、前記方法は、前記第１レベルの前記受信キャッシュから第２の階層レベルの受信キャッシュに、前記オブジェクトのコピーを要求するメッセージを送信するステップをさらに含む方法。

【請求項９】　請求項４に記載の方法において、前記ポインタは、前記オブジェクトのディレクトリリストを記憶しているディレクトリキャッシュのネットワークアドレスを指す方法。

【請求項１０】　請求項９に記載の方法であって、前記受信キャッシュから前記ポインタによって示されたアドレスをもつ前記ディレクトリキャッシュに、オブジェクトリクエストメッセージを送信するステップをさらに含む方法。

【請求項１１】　請求項１０に記載の方法であって、前記ディレクトリキャッシュから前記受信キャッシュに、

前記オブジェクトの前記ディレクトリリスト上のオブジェクトキャッシュのネットワークアドレスを含むメッセージを送信するステップをさらに含む方法。

【請求項１２】　請求項１０に記載の方法であって、前記ディレクトリキャッシュから前記オブジェクトの前記ディレクトリリストに掲載されたアドレスを持つオブジェクトキャッシュに、前記オブジェクトキャッシュが前記オブジェクトのコピーを記憶しているかどうかを示すメッセージを受信キャッシュに送信するよう前記オブジェクトキャッシュに要求するメッセージを送信するステップをさらに含む方法。

【請求項１３】　請求項１０に記載の方法において、複数のディレクトリリストが複数のディレクトリキャッシュ上に記憶されており、各ディレクトリリストは、オブジェクトのネットワークアドレスと、前記オブジェクトのコピーを記憶していると称される少なくとも１つのオブジェクトキャッシュのネットワークアドレスとを含み、前記方法は、

前記ディレクトリリストからα個のオブジェクトキャッシュのネットワークアドレスを選択するステップと、

前記ディレクトリキャッシュから選択された各オブジェクトキャッシュに、オブジェクトキャッシュが前記オブジェクトのコピーを記憶しているかどうかを示すメッセージを前記受信キャッシュに送信するよう要求するメッセージを送信するステップと、

前記選択された各オブジェクトキャッシュから前記受信キャッシュに、前記オブジェクトキャッシュが前記オブジェクトのコピーを記憶しているかどうかを示すメッセージを送信するステップと、

前記オブジェクトのコピーを記憶していることを示す、前記受信キャッシュが受信した最初のメッセージを送信したオブジェクトキャッシュを、主要オブジェクトキャッシュとして特定するステップと、

前記受信キャッシュから前記主要オブジェクトキャッシュに、前記オブジェクトのコピーを前記受信キャッシュに送信するよう要求するメッセージを送信するステップと、をさらに含む方法。

【請求項１４】　請求項１０に記載の方法において、複数のディレクトリリストが複数のディレクトリキャッシュ上に記憶されており、各ディレクトリリストは、オブジェクトのネットワークアドレスと、前記オブジェクトのコピーを記憶していると称される少なくとも１つのオブジェクトキャッシュのネットワークアドレスとを含み、前記方法は、

前記ディレクトリキャッシュから前記受信キャッシュに、前記オブジェクトのディレクトリリスト上のオブジェクトキャッシュのアドレスを含むメッセージを送信するステップと、

前記ディレクトリリストからα個のオブジェクトキャッシュのネットワークアドレスを選択するステップと、

前記受信キャッシュから選択された各オブジェクトキャッシュに、そのオブジェクトキャッシュが前記オブジェクトのコピーを記憶しているかどうかを示すメッセージを前記受信キャッシュに送信するよう要求するメッセージを送信するステップと、

前記選択された各オブジェクトキャッシュから前記受信キャッシュに、そのオブジェクトキャッシュが前記オブジェクトのコピーを記憶しているかどうかを示すメッセージを送信するステップと、

前記オブジェクトのコピーを記憶していることを示す、前記受信キャッシュが受信した最初のメッセージを送信したオブジェクトキャッシュを、主要オブジェクトキャッシュに特定するステップと、

前記受信キャッシュから前記主要オブジェクトキャッシュに、前記オブジェクトのコピーを前記受信キャッシュに送信するよう要求するメッセージを送信するステップと、をさらに含む方法。

【請求項１５】　ネットワーク上のスケーラブルな分散型キャッシングシステムであって、

データオブジェクトのコピーを記憶する手段と、

前記オブジェクトのコピーが記憶されているネットワーク位置を特定するディレクトリリストを記憶する手段と、

前記オブジェクトの前記ディレクトリリストが記憶されているネットワーク位置を求める手段と、

前記オブジェクトのコピーを要求する場所を選択する手段と、

前記オブジェクトのコピーを入手する手段と、を含むシステム。

【請求項１６】　請求項１５に記載のシステムであって、オブジェクトが記憶される新しいネットワーク位置を前記オブジェクトの前記ディレクトリリストに加える手段をさらに含むシステム。

【請求項１７】　請求項１５に記載のシステムであって、あるオブジェクトの前記ディレクトリリストからネットワーク位置を削除する手段をさらに含むシステム。

【請求項１８】　請求項１５に記載のシステムであって、記憶されたオブジェクトのコピーを削除する手段をさらに含むシステム。

【請求項１９】　請求項１５に記載のシステムであって、あるオブジェクトのディレクトリリストを削除する手段をさらに含むシステム。

【請求項２０】　請求項１５に記載のシステムであって、前記オブジェクトのオリジナル情報源からオブジェクトのコピーを入手する手段をさらに含むシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はネットワーク上のオブジェクトのキャッシングに関し、より特定的には、スケールラブルな分散型ディレクトリを用いて、キャッシ

ュされたオブジェクトを管理する方法に関する。

【０００２】

【背景技術】データオブジェクトのキャッシングは、ネットワーク情報サービスのスケーラビリティの向上に利用される。キャッシングは、大量のデータオブジェクトがネットワークに加えられる場合に発生しうるネットワーク上のサービスの劣化を抑える助けとなる。キャッシングを用いない場合、ユーザがネットワークを通じてデータオブジェクトをリクエストすると、そのデータオブジェクトの情報源にオブジェクトのコピーの送信が要求される。遠隔サイトやデータオブジェクトの数が多い大規模なネットワークでは、この要求を処理するのに長時間かかるか、またはネットワークの状態、ネットワークトラフィック、およびデータオブジェクト情報源の状態によってまったく処理不能となる場合がある。キャッシングを用いるシステムは、オブジェクト情報源以外の最低一箇所、多くの場合はリクエスト元の近くの位置に、データオブジェクトのコピーを記憶する。オブジェクトの別情報源が最低１つ存在すること、およびその別情報源とリクエスト元のユーザとの状況がおそらくはより好都合なために、リクエスト元のユーザにコピーを迅速かつ正しく配信してサービスの向上を計ることができる。

【０００３】オブジェクトキャッシングが使用されるネットワークの一例として、インターネットがある。インターネット上でキャッシングを用いるには、オリジナル情報源のサーバ以外にインターネットに接続された最低１つのサーバ上に、（ウェブサイト等からの）ハイパーテキストファイルのコピーをキャッシュする。例えば、日本のサーバ上のある人気のあるファイルを米国のクライアントがリクエストする度に、インターネット上でワールドワイドウェブを介してそのファイルを取り寄せる代わりに、米国内でそのファイルをコピーしてキャッシュサーバに記憶しておくことができる。その後、そのファイルを米国の他のクライアントがリクエストする場合は、日本のサーバではなく米国内のキャッシュサーバからそのファイルを入手することができる。この利点は、リクエストおよびコピーのどちらも、リクエスト元と日本のオリジナル情報源のサーバとの間よりもキャッシュサーバとの間のほうが、インターネット中で通過するルータおよびノード（中継点）の数が少なくてすむことである。このため、コピー入手までの遅れが少なくなり、かつリクエスト元のユーザがコピーを正しく受信できる可能性が高くなる。こうして、米国のクライアントが日本のファイルを入手する費用および待ち時間（遅れ）を大幅に低減することができる。

【０００４】キャッシングを採用するシステムは、３つの機能上の構成要素からなることが理解できる。第１に、キャッシュは、ユーザによってリクエストされたデータオブジェクトのキャッシュされたコピーの場所を決定する検索システムを含む。第２に、キャッシングを採用するシステムは、リクエストされたオブジェクトの１つ以上のキャッシュされたコピーの場所を検索システムが突き止めた場合に、どのコピーを取り出すのかを決定する決定システムを含む。最後に、維持システムは、古い期限切れのオブジェクトのコピーをキャッシュから取り除き、キャッシングによって記憶された情報の正確さを全体的に維持する助けをする。

【０００５】図１に従来のキャッシングシステムの階層構造を示す。クライアント１０１はキャッシュサーバＡ１０２にデータオブジェクトをリクエストする。以下、「キャッシュ」はキャッシュサーバをさすものとする。キャッシュＡ１０２（クライアントからオブジェクトのリクエストを受信するキャッシュ）は、一般にはネットワーク的にクライアントの「近くに」位置する。つまり、キャッシュＡとクライアントとは、直接またはネットワークを介して迅速かつ信頼性をもって、または少なくともクライアント１０１がキャッシュＡ以外のサーバと通信するよりは迅速かつ信頼性をもって、互いに通信しあえる。ネットワーク的にクライアントの「近くに」位置するキャッシュサーバは、ネットワーク的に「近くない」他のサーバよりも物理的には離れている場合があることを理解されたい。

【０００６】クライアントのリクエストは、クライアント１０１とキャッシュＡ１０２とを接続するネットワーク（図示せず）を介して送信できる。もしキャッシュＡがオブジェクトのコピーを持っていれば、キャッシュＡはそれをクライアント１０１に送信する。もしリクエストされたデータを持っていなければ、キャッシュＡ１０２はキャッシュされたコピーを検索しなければならない。この場合、キャッシュＡ１０２はまず、階層レベルが同じ近隣キャッシュ群、つまりキャッシュＢ１０３、キャッシュＣ１０４、およびキャッシュＤ１０５にリクエストを送信する。キャッシュＡ１０２、キャッシュＢ１０３、キャッシュＣ１０４、およびキャッシュＤ１０５は、ネットワーク（図示せず）で接続できる。各キャッシュは、応答中のキャッシュがリクエストされたオブジェクトのコピーを持っているかどうかを知らせるメッセージをキャッシュＡ１０２に送って、キャッシュＡ１０２からのリクエストに応じる。同階層レベルのキャッシュ群がリクエストされたオブジェクトを持っていなければ、キャッシュＡ１０２は次の階層レベルのキャッシュＥ１０６にオブジェクトのリクエストを送信する。キャッシュＡ１０２は、ネットワーク（図示せず）によってキャッシュＥ１０６に接続できる。同様に、もしキャッシュＥ１０６がオブジェクトを持っていなければ、キャッシュＥ１０６は同じ階層レベルのキャッシュ群、つまりキャッシュＦ１０７およびキャッシュＧ１０８をポーリングする。キャッシュＥ１０６、キャッシュＦ１０７、およびキャッシュＧ１０８はネットワーク（図示せず）で接続できる。これらのキャッシュ群のどれもオブ

ジェクトを持っていない場合は、オブジェクトのオリジナルバージョンが存在するサーバ１０９にメッセージが送られる。サーバ１０９に送信されるメッセージは、リクエストされたオブジェクトのコピーをクライアント１０１に送信するようサーバ１０９に要求する。いずれか１つのキャッシュがオブジェクトのコピーを持っているというメッセージを送信する場合は、そのキャッシュがクライアント１０１にオブジェクトのコピーを送信するよう要求される。一方、１つ以上のキャッシュ群が、オブジェクトのコピーを持っているというメッセージを送信する場合は、決定関数を実行して、どのキャッシュがオブジェクトのコピーをクライアント１０１に送信するかを選択しなければならない。

【０００７】図２～図４は公知の決定システムを示す。クライアント２０１はキャッシュＡ２０２にオブジェクトのリクエスト（図２では“ｒｅｑｕｅｓｔ（１）”で示す）を送信する。キャッシュＡ２０２はコピーを持っているかどうかを判断する。もし持っていれば、キャッシュＡ２０２はクライアント２０１にコピーを送信する。もし持っていなければ、キャッシュＡ２０２はすべての近隣キャッシュ群２０３～２０７に、“ＵＤＰ＿ｐｉｎｇ＿ｒｅｑ（２）”で示すＵＤＰ　ｐｉｎｇリクエストと呼ばれるリクエストを送信する。各メッセージの後の括弧内の数字（“ｒｅｑｕｅｓｔ（１）”、“ＵＤＰ＿ｐｉｎｇ＿ｒｅｑ（２）”等）は、メッセージが送信される順番を示す。

【０００８】図３に示すように、各近隣キャッシュは、もしオブジェクトのコピーを持っていればＵＤＰ　ｐｉｎｇヒットメッセージ（図３では“ＵＤＰ＿ｐｉｎｇ＿ｈｉｔ”で示す）で応答し、もし持っていなければＵＤＰ＿ｐｉｎｇミスメッセージ（図３では“ＵＤＰ＿ｐｉｎｇ＿ｍｉｓｓ”で示す）で応答する。ＵＤＰ　ｐｉｎｇヒットメッセージがまったく受信されない場合は、リクエストされたオブジェクトのコピーの送信はオリジナル情報源のサーバに依頼される。キャッシュＡ３０２がＵＤＰ　ｐｉｎｇヒットメッセージを１つだけ受信する場合は、キャッシュＡ３０２はコピーを持っている唯一のキャッシュにそのオブジェクトのコピーをクライアント３０１に送信するよう要求するメッセージを送信する。図３に示すように１つ以上のＵＤＰ　ｐｉｎｇヒットメッセージを受信する場合は、キャッシュＡ３０２は、どのキャッシュにオブジェクトコピーをクライアント３０１へ送信するよう依頼すべきかを決定しなければならない。この先行技術では、図４に示すように、キャッシュＡ４０２は、ＵＤＰ　ｐｉｎｇヒットメッセージを最初に受信したキャッシュ、つまりキャッシュＥ４０３（図４）に、オブジェクトのコピーを要求するメッセージ（“ＲＥＱＵＥＳＴ（１）”）を送信する。この決定関数は、ネットワーク的にキャッシュＡ３０２に（従ってクライアント３０１に）「最も近い」コピーを入手するように設計されている。このため、リクエストされたオブジェクトのコピーを、最小の遅れと最大の信頼性で入手することができる。

【０００９】コピーはキャッシュＥ４０３からキャッシュＡ４０２へ送られ（ＣＯＰＹ（２））、ネットワーク的にはキャッシュＥよりもクライアント４０１に「より近い」キャッシュＡ４０２にオブジェクトのコピーが記憶される。この結果、このオブジェクトに対するクライアント４０１からのこれ以降のリクエストでは、キャッシュＡ４０２以外の検索は不要となる。次に、オブジェクトのコピーはキャッシュＡ４０２からクライアント４０１へ送信される（ＣＯＰＹ（３））。先行技術の他の実施形態では、リクエストは、キャッシュＡ４０２をバイパスして、キャッシュＥ４０３にオブジェクトのコピーを直接クライアント４０１に送ることを要求する。

【００１０】キャッシングを採用したシステムのこの先行技術の実施形態は、待ち時間（リクエストとオブジェクトのコピーの受信との間の遅れ）を短縮し、比較的データオブジェクトの数が少ない小規模なネットワーク中では、リクエストされたオブジェクトのコピー入手の信頼性を向上させる。しかし大規模なネットワークでは、かかる公知の方法はスケーラビリティ（拡張性）が低い。例えばオブジェクトに対するクライアントからのリクエストを受信するキャッシュ（以後、「受信キャッシュ」と称する）がＮ個の近隣キャッシュを持つキャッシングのシステムを考える。これらのキャッシュ群は、リクエストされたオブジェクトのコピーを保有するメッセージ以外に、各クライアントリクエストごとに２Ｎ＋１個のメッセージを作成する。つまり、各近隣キャッシュごとに２つのメッセージ（ＵＤＰｐｉｎｇリクエストと、ＵＤＰ　ｐｉｎｇヒット応答またはＵＤＰ　ｐｉｎｇミス応答のいずれか）と、クライアントからのリクエストメッセージと、である。このように大量のメッセージは、データオブジェクト数の多い大規模ネットワークでは大変な負担となる。１つのキャッシュが追加されるごとに、キャッシュシステムが作成するメッセージ数が２つずつ増加する。

【００１１】この問題を解決するための公知の方法は、ディレクトリ（例えば、データオブジェクトを識別する情報と、そのキャッシュ場所のリスト）を使用して、データオブジェクトのＩＤを該オブジェクトが記憶されているキャッシュ場所に対応づけることである。これにより全キャッシュをフルに検索する必要がなくなり、キャッシングシステムがネットワークに課すメッセージトラフィック量を少なくできる。例えば、メッセージをすべての近隣キャッシュ群へ送信する代わりに、１つのリクエストをディレクトリをもつ中央キャッシュへ送信して、中央キャッシュにリクエスト元に一番近い、リクエストされたコピーの場所を問い合わせる。

【００１２】ディレクトリを使用すると、キャッシュさ

れたオブジェクトのコピーの場所をつきとめて入手するのに必要なトラフィック量を低減できるが、キャッシュコヒーレンスの問題が生じる。つまり、データオブジェクトが変更されると、維持システムによってそのデータオブジェクトの全コピーを変更し、かつ全ディレクトリを更新しなければならない。分散型コヒーレンスプロトコルを用いてオブジェクトとディレクトリとのコヒーレンスを維持し、ネットワークを介して送信されるメッセージを用いて古いオブジェクトのコピーを無効にし、ディレクトリのエントリを更新する。しかし、このメッセージトラフィックはネットワーク、特に変更頻度の高い多数のオブジェクトをもつ大規模ネットワークに負担をかけてしまう。

【００１３】オブジェクトのコピーが古くなる（つまりオリジナルのデータオブジェクトが変更される）と、無効コピーは無効化されてキャッシュ群から除去または排除される。さらに、キャッシュされたコピーをさすディレクトリが更新される。これは従来のいくつかのシステムでは、各コピーに生存時間（ＴＴＬ）パラメータを割り当てることによって行われる。ＴＴＬパラメータは、記憶されたデータオブジェクトのコピーが期限切れして削除されるべき日時を特定する。ＴＴＬが到来すると、コピーはキャッシングシステムから排除される（つまりキャッシュから削除される）。ＴＴＬが期限切れする前に古くなったコピーを排除しようとすれば、従来の大規模なキャッシングシステム中ではトラフィックの負担が大きすぎて実現できない。従って、従来のシステムはＴＴＬパラメータに基づいて弱いコヒーレンスしか維持できない。コヒーレンスが弱いのは、キャッシュ化されてからすぐに変更されたオブジェクトのキャッシュされたコピーが、ＴＴＬが切れるまで、古くなってから長時間にわたって利用可能なまま残留するためである。この問題は、オブジェクトのコピーおよびディレクトリ中のそのエントリを削除するメッセージをばらまくことによって解決できるが、これによりトラフィックでネットワークにさらに負担をかけてしまう。

【００１４】ロッキングおよび承認も、従来のキャッシングシステムのスケーラビリティを下げる原因である。従来のシステム中でディレクトリが更新される場合、変更中はディレクトリがロック（つまりアクセス阻害）されなければならない。このため更新中はディレクトリは利用不可となる。また、更新の受信を示すために承認メッセージが送信されなければならず、ネットワークの一層の負担となる。従って、ロッキングまたは承認が不要なスケーラブルなディレクトリベースのキャッシングシステムであれば、従来のキャッシュシステムよりもネットワークの性能をより効率的かつ効果的に改善することができる。

【００１５】

【課題を解決するための手段】本発明は、ネットワーク上でロッキングおよび承認なしでオブジェクトキャッシングを有利かつ効率的に管理する、スケーラブルな分配型ディレクトリベースのキャッシュ方式システムを提供する。このシステムは、キャッシュネットワークトラフィックが少なく、メモリおよび処理電力が従来のキャッシュシステムより少なくてすむ。

【００１６】本発明に従えば、オブジェクトに対するユーザリクエストは受信キャッシュによって受信される。受信キャッシュは、リクエストされたオブジェクトのネットワークアドレスを入力として用い、出力としてポインタを与えるロケータ関数を実行する。一実施形態では、ポインタは、オブジェクトのディレクトリリストを記憶するディレクトリキャッシュのネットワークアドレスを指す。オブジェクトのディレクトリリストは、オブジェクトのネットワークアドレスと、オブジェクトのコピーを記憶していると称されるオブジェクトキャッシュ群のネットワークアドレスとを含む。

【００１７】受信キャッシュはディレクトリキャッシュにオブジェクトリクエストメッセージを送信し、ディレクトリキャッシュはディレクトリリストの掲載されたオブジェクトキャッシュ群をポーリングする。これに応答して、オブジェクトキャッシュ群は、各オブジェクトキャッシュがオブジェクトのコピーを記憶しているかどうかを示すメッセージを受信キャッシュに送信する。

【００１８】受信キャッシュは、オブジェクトのコピーを記憶していることを示すメッセージであって、受信キャッシュが受信した最初のものを送信したオブジェクトキャッシュに、オブジェクトリクエストメッセージを送信する。これに応答して、送信元のオブジェクトキャッシュは、受信キャッシュにオブジェクトのコピーを送信し、受信キャッシュはコピーを記憶して、ユーザにコピーを転送する。

【００１９】ディレクトリリストはその後、受信キャッシュのネットワークアドレスを追加して更新される。古くなったキャッシュされたオブジェクトのコピーは、オブジェクトのオリジナル情報源のサーバからキャッシュに新しいコピーが送信されるごとに、分散方式で削除される。キャッシュされたコピーのコヒーレンスは、生存時間（ｔｉｍｅ－ｔｏ－ｌｉｖｅ）パラメータを各コピーに関連づけることによってさらに向上できる。生存時間パラメータはまた、オブジェクトのディレクトリリスト上のオブジェクトキャッシュのアドレスに関連づけられる。

【００２０】本発明の分散型機能によって、本発明はスケーラブルかつ効率的となり、データオブジェクトのコピーをキャッシュし、キャッシュされたコピーを迅速かつ経済的にユーザに与えることができる。

【００２１】

【発明の実施の形態】本発明は、ネットワーク上でキャッシュ群を用いて実現されるスケーラブル分散型キャッ

シングシステムを提供する。キャッシュはサーバであり、プロセッサとコンピュータにより読み出し可能なメモリとネットワークに接続されるポートとの間を接続するデータバスを含む。本発明に従うキャッシュは、データオブジェクトのコピーを記憶する。キャッシュはまた、オブジェクトのコピーを記憶するキャッシュ群のアドレスを掲載するオブジェクトのディレクトリリストを記憶（または該リストのみを記憶）する。キャッシュはまた、オブジェクトのネットワークアドレスに基づいて、オブジェクトのディレクトリリストを記憶するキャッシュのネットワークアドレスの位置を求める際に有用なハッシュロケータ関数を実現できる。本明細書において、「オブジェクトのネットワークアドレス」はオリジナルオブジェクトの情報源のネットワークアドレス上の位置を特定可能なあらゆるものをさす。オブジェクトのネットワークアドレスの一例は、オブジェクトのオリジナル情報源におけるユニフォーム・リソース・ロケータ（ＵＲＬ）である。「ネットワークアドレス」をサーバ＊（キャッシュ等）に関して用いる際は、ネットワーク上でサーバ位置を特定可能なあらゆるものをさす。「ネットワークアドレス」をあるオブジェクトのキャッシュされたコピーに関して用いる際は、ネットワーク上でオブジェクトのキャッシュされたコピーの位置を求めることが可能なあらゆるものをさす。オブジェクトのキャッシュされたコピーのネットワークアドレスの一例はＵＲＬである。

【００２２】本発明に従えば、各キャッシュされたデータオブジェクトはディレクトリリストを有する。オブジェクトのディレクトリリスト上の各アドレスは、オブジェクトのコピーを記憶するキャッシュ（オブジェクトキャッシュ）のアドレスである。好適な実施形態では、ディレクトリリスト群は複数のキャッシュ（ディレクトリキャッシュ）間に分散される。本発明に従うディレクトリリストのデータ構造の一実施例を以下に示す。

【００２３】

【表１】

```
Directory_list
        {
         directory_object_address
                  /*リストオブジェクトのネットワークアドレス*/
         {
                  /*リストオブジェクトのコピーが記憶されている
                    ５つのオブジェクトキャッシュ群のアドレス*/
                  object_cache_address_1
                  object_cache_address_2
                  object_cache_address_3
                  object_cache_address_4
                  object_cache_address_5
         }
        }
```

クライアントからオブジェクトリクエストを受信する各キャッシュは、ハッシュロケータ関数を備える。ハッシュロケータ関数の入力はオブジェクトのネットワークアドレスであり、出力はリクエストされたオブジェクトのディレクトリリストが記憶されているディレクトリキャッシュのアドレス（またはそのポインタ）である。こうして、受信キャッシュはキャッシュされた任意のオブジェクトについてディレクトリキャッシュの位置を求めることができる。リクエストされたオブジェクトにディレクトリリストがない（つまりロケータ関数がどのディレクトリキャッシュも指さない）場合は、リクエストはオリジナル情報源のサーバに送られ、リクエストされたオブジェクトを受信キャッシュに送信させる。受信キャッシュはリクエストされたオブジェクトのコピーを記憶し、そのコピーをクライアントに送信する。その後、受信キャッシュは、オブジェクトキャッシュとして受信キャッシュアドレスを含むオブジェクトのディレクトリリストを開始させる。一実施形態では、受信キャッシュはオブジェクトのディレクトリリストも記憶する。この例からわかるように、キャッシュは、受信キャッシュ（クライアントのリクエストを受信）、ディレクトリキャッシュ（オブジェクトのディレクトリリストを記憶）、およびオブジェクトキャッシュ（オブジェクトのコピーを記憶）の機能を同時に行うことができる。

【００２４】図５は、ネットワークアドレスＹをもつデータオブジェクトへのリクエスト（図５では“ＧＥＴ＿ｒｅｑｕｅｓｔ（Ｙ）”で示す）をクライアント５０２から受信する受信キャッシュであるキャッシュＡ５０１を示す。受信キャッシュＡ５０１は、アドレスＹを入力として用いてハッシュロケータ関数Ｌを実行し、ディレクトリキャッシュアドレスＣを出力として得る。これはつまり、キャッシュＣ５０３がオブジェクトＹのディレクトリリストを記憶していることを示す。オブジェクトＹのディレクトリリストのデータ構造の一実施例を以下に示す。

【００２５】

【表２】

```
Directory_list
        {
         Y    /*オブジェクトＹのネットワークアドレス*/
         {
              /*それぞれディレクトリリストオブジェクトＹの
                コピーを記憶しているオブジェクトキャッシュ
                Ｂ，Ｄ，Ｅ，およびＦのアドレス*/
              B
              D
              E
              F
         }
        }
```

上記のデータ構造は、データオブジェクトＹのコピーがオブジェクトキャッシュＢ５０４，Ｄ５０５，Ｅ５０６，およびＦ５０７に記憶されていることを示す。受信キャッシュＡ６０１（図６）は、データオブジェクトをリクエストするメッセージ（図６では“ＬＯＯＫＵＰ”で示す）をディレクトリキャッシュＣ６０２に送信する。図６に示す実施形態では、ディレクトリキャッシュＣ６０２は受信キャッシュＡ６０１に、ディレクトリリストからのオブジェクトキャッシュアドレス群を含むメッセージ（図６では“ＬＩＳＴ［Ｂ，Ｄ，Ｅ，Ｆ］”で示す）を送信する。キャッシュＡ６０１は、このリストを用いてオブジェクトキャッシュ群をポーリングし、その中からオブジェクトＹを受信するキャッシュを１つ選択する。本発明の他の実施形態では、ディレクトリキャッシュＣ６０２はオブジェクトキャッシュのアドレス群を含むメッセージを受信キャッシュＡ６０１に送信せず、代わりにディレクトリキャッシュＣ６０２自身がオブジェクトキャッシュ群（図６の実施形態ではＢ－６０３，Ｄ－６０４，Ｅ－６０５，およびＦ－６０６で示す）をポーリングし、各オブジェクトキャッシュにＵＤＰ　ｐｉｎｇヒットまたはミスメッセージを受信キャッシュＡ６０１に送信するよう依頼する。その後、受信キャッシュＡ６０１は、どのオブジェクトキャッシュからオブジェクトＹのコピーを要求するかを決定し、選択したオブジェクトキャッシュにコピーの送信を要求するメッセージを送る。

【００２６】本発明は、所与のオブジェクトのキャッシュされたコピーの位置を求めるために非分散型ディレクトリを使用する従来のディレクトリシステムよりも効率的である。ある従来のシステムでは、キャッシュされたオブジェクトに対するリクエストがあるたびに、１つのサーバに存在する１つの非分散型ディレクトリに問い合わせが行われる。この単独のディレクトリは、リクエストされたオブジェクトが記憶されているキャッシュの最低１つのアドレスを返送する。この従来のシステムの欠点は、単独サーバがキャッシュされたオブジェクト群に対する全リクエストのボトルネックとなり、リクエストされたオブジェクトを迅速かつ信頼性をもって返信するシステムの性能の向上というキャッシングの目的を妨げることである。クライアントと単独ディレクトリサーバとの間のリクエストメッセージおよび応答メッセージのトラフィックによって、またディレクトリの正確さを維持するための維持トラフィックによって、ネットワークに過度の負担がかけられる。単独ディレクトリサーバに対する負担が重くなってサーバの実行速度が遅くなり、サーバを他の（ディレクトリ目的以外の）タスクに使用できなくなる場合がある。さらに、非分散型サーバは、サーバ上でかなり大量のメモリリソースをとるためにサイズが大きくなり、探索速度が落ちてプロセッサの処理時間が長くなる場合がある。また単独ディレクトリサーバは、キャッシングシステムの単独故障点でもある。もし単独ディレクトリサーバが故障すれば、キャッシングシステム全体が動作不能になり、キャッシュされたコピーをオブジェクトキャッシュ群に記憶するのに使用されたすべてのメモリ資源が無駄になってしまう。

【００２７】従来のシステムの問題の一部を解決する明らかな方法は、ネットワーク上の複数の異なる場所（すなわちネットワークアドレスの異なる複数の異なるサーバ上）に、非分散型ディレクトリの複数のコピーを置くことである。これにより、クライアントはオブジェクトキャッシュのアドレスを入手するのに、多数の非分散型ディレクトリの１つに問い合わせができるから、単独ディレクトリサーバを有することによって生じるボトルネックおよび単独故障点の影響を減じることができる。これはディレクトリ探索活動の負担を分散させる助けとなる。また、１つのディレクトリサーバが故障しても、別のサーバに問い合わせを行うことができる。しかし、この解決方法では、ディレクトリ群の正確さを維持する必要があるため、ネットワーク上のキャッシングシステムトラフィックによる負担を悪化させてしまう。従って、キャッシュされたコピーの状態が変更される（キャッシュされたコピーが古くなったり、新たなオブジェクトがキャッシュ化される等）と、全ディレクトリサーバに更新メッセージの送信が要求されるため、承認メッセージ

の形式でさらに大量のトラフィックを発生させてしまう。これにより、かなり大量のメッセージトラフィックを発生させてネットワークに負担をかけてしまうという欠点がある。また、単独非分散型ディレクトリの場合と同じく、従来のシステムにおける１つ以上のディレクトリをもつ非分散型ディレクトリ群は、更新中はロックされなければならず、その間は使用できない。オブジェクトの状態が頻繁に変更される大規模なシステムでは、更新およびロッキングは非常に頻繁に発生するため、探索ディレクトリの利用可能性を大幅に下げてしまう。また、大規模な非分散型ディレクトリの複数のコピーをもつことにより、メモリおよびプロセッサ資源を奪ってしまうという問題が悪化する可能性がある。

【００２８】本発明はこのような問題を解決し、従来のシステムより効率よくかつ経済的に使用および維持可能な、キャッシングシステム用の分散型ディレクトリを提供する。本発明に従ってディレクトリをディレクトリリスト群に分割して、これらのリストを多数のサーバに分散させることによって、従来のシステムよりも多数のサーバにディレクトリ参照の負担を分散することができる。これによりボトルネックおよび単独故障点の問題がかなり低減または解消される。非分散型ディレクトリキャッシングシステムのディレクトリサーバと比べて、各ディレクトリサーバ上のディレクトリの記憶に必要なメモリ資源が大幅に少なくなるため、サーバはディレクトリデータ以外のデータを記憶できる。さらに、このようにディレクトリ探索の負担が有利に分散されるため、任意の１つのサーバが吸収するプロセッサ時間がはるかに少なく、ディレクトリサーバが探索リクエストに迅速かつ効率的に応答できる。本発明の他の利点は、ディレクトリ維持に必要なトラフィックレベルが低く、従来のシステムに比べてネットワーク上の負担が少ないことである。本発明は承認メッセージを使用せず、有利には更新時にディレクトリデータをロックせず、かつ承認メッセージを使用しない。

【００２９】検索システムは、リクエストされたオブジェクトのコピーが記憶されている場所がもしあれば、その場所を求める。上述したように、本発明に従えば、キャッシュはクライアントからオブジェクトの入手を求めるリクエスト（“ＧＥＴ　ｒｅｑｕｅｓｔ”と称する）を受信する。図５は、クライアント５０２が入手リクエストをキャッシュＡ５０１に送信するキャッシングシステムの一実施形態を示す。リクエストを受信すると、受信キャッシュであるキャッシュＡ５０１はまず、要求されたオブジェクトを自身が記憶しているかどうかを判断する。キャッシュＡ５０１がリクエストされたオブジェクトを持っている場合は、該オブジェクトがクライアント５０２に送信される。持っていない場合は、キャッシュＡ５０１はどこか他の場所でオブジェクトの位置を求めなければならない。

【００３０】リクエストされたオブジェクトを持っていない場合、キャッシュＡ５０１はディレクトリリストロケータ関数Ｌ５０８を実行する。上述したように、ディレクトリリストロケータ関数Ｌ５０８は、リクエストされたデータオブジェクトのネットワークアドレスを入力として使用して、該オブジェクトのディレクトリリストを記憶するキャッシュのアドレス（またはそのポインタ）を出力する。このため、キャッシュＡ５０１がクライアント５０２からデータオブジェクトＹに対するリクエストを受信し、キャッシュＡ５０１がオブジェクトＹを記憶していないと判断すると、キャッシュＡはＬ５０８を実行して、Ｙのディレクトリリストの位置を求める。図５に示すＬ（Ｙ）＝Ｃは、キャッシュＣ５０３がオブジェクトＹのディレクトリリストを記憶していることを示す。

【００３１】本発明のある実施形態においては、ロケータ関数はネットワークアドレスを示さない。これは、ロケータ関数を実行する受信キャッシュの論理上、到達可能な範囲内のキャッシュにオブジェクトのコピーがない、またはどのキャッシュにもオブジェクトのコピーが記憶されていないことを意味する。受信キャッシュの論理範囲内のキャッシュにオブジェクトのコピーが記憶されていない本発明の実施形態では、キャッシュ群は論理レベルの階層中で論理レベルごとにグループ分けされる。第１の論理レベルのキャッシュ群の１つは、同レベルの他のキャッシュに記憶されているオブジェクトのコピーを検索して入手できる。第２レベルのキャッシュに記憶されているオブジェクトのコピーを入手するには、第２レベルの受信キャッシュにオブジェクトリクエストメッセージが送信され、第２レベルの受信キャッシュは第２レベルのキャッシュからオブジェクトのコピーの検索を開始し、該コピーの入手を試みる。コピーを入手すると、そのコピーは第１レベルの受信キャッシュへ送られる。本実施形態では、ロケータ関数が、第１レベルの受信キャッシュが論理上利用可能なキャッシュのどれもがオブジェクトのディレクトリリストを記憶していないことを示すポインタを返送する場合は、第１レベルの受信キャッシュは第２レベルの受信キャッシュへのオブジェクトリクエストの送信を選択する。

【００３２】また、本発明の他の実施形態では、ロケータ関数から返送されたポインタが、受信キャッシュが論理上利用可能なキャッシュのどれもがオブジェクトのディレクトリリストを記憶していないことを示す場合は、受信キャッシュはオブジェクトのオリジナルバージョンを記憶しているサーバにオブジェクトのコピーをリクエストすることを選択する。

【００３３】このように、ハッシュ関数Ｌ５０８は、オブジェクトのネットワークアドレスを入力として使用し、出力としてポインタを与えて、該ポインタが示す情報が本発明のこれ以降の行動を決定する関数を含む。ポ

インタがオブジェクトのディレクトリリストを記憶しているディレクトリキャッシュのネットワークアドレスを示すと、オブジェクトリクエストは受信キャッシュからディレクトリキャッシュへ送られる。一方、ポインタがネットワークアドレスをまったく示さない場合、または論理的に利用可能なキャッシュのネットワークアドレス（同じ階層レベルのキャッシュのアドレス等）を示さない場合は、オブジェクトリクエストメッセージは他の階層レベルの受信キャッシュに送られるか、またはオブジェクトのオリジナル情報源サーバへ送られる。一実施形態では、ロケータ関数のポインタをディレクトリキャッシュのネットワークアドレス等の情報と相関させるテーブルが、システム管理者によって手動で受信キャッシュに入力される。他の実施形態では、ポインタテーブルを更新する受信キャッシュにメッセージが送られる。例えば、新たにキャッシュされたオブジェクトＸについて新たなディレクトリリストがディレクトリキャッシュＺにおいて開始されると、オブジェクトＸのネットワークアドレス、ディレクトリキャッシュＺのネットワークアドレス、およびポインタテーブルが更新されるべきであることを示すフラグを含むメッセージが、受信キャッシュに送信される。受信キャッシュはこのメッセージを受信し、オブジェクトのネットワークアドレスに対してハッシュ関数を実行し、得られたポインタ値をキャッシュＺのネットワークアドレスに相関させる。このように、オブジェクトＸに対するリクエストを受信すると、受信キャッシュはＸに対してハッシュロケータ関数を行い、ポインタを入手し、そのポインタをキャッシュＺのネットワークアドレスに相関させる。その後、受信キャッシュは、本発明に従ってキャッシュＺに探索リクエストを送る。

【００３４】図５に示す例では、キャッシュＡ５０１はオブジェクトＹの入手リクエストを受信している。キャッシュＡ５０１は、それ自身はオブジェクトＹを記憶していないと判断し、Ｙのネットワークアドレスに対してロケータ関数Ｌ５０８を実行している。本例では、Ｌ（Ｙ）＝Ｃであり、キャッシュＣ５０３がオブジェクトＹのディレクトリリストを記憶していることを示す。

【００３５】図６は、図５においてディレクトリリストロケータ関数Ｌ５０８が、キャッシュＣがオブジェクトＹのディレクトリキャッシュを記憶していると特定した後の様子を示す図である。キャッシュＡ６０１はキャッシュＣ６０２に“探索リクエスト（ＬＯＯＫＵＰ　ｒｅｑｕｅｓｔ）”を送信する。一実施形態では、探索リクエストは、リクエストされたオブジェクトＹのアドレスと、リクエスト元のキャッシュつまりキャッシュＡ６０１のアドレスとを含む。キャッシュＣ６０２は、リクエストされたオブジェクトＹのディレクトリリストキャッシュのアドレス群をキャッシュＡ６０１に返送する。キャッシュＡ６０１は決定システムを実行して、この中からオブジェクトのコピーを要求するキャッシュを選択する。他の実施形態では、キャッシュＣ６０２はキャッシュＡ６０１にディレクトリリストを送信せず、この代わりにキャッシュＣ６０２は、リクエストされたオブジェクトのディレクトリリスト上のオブジェクトキャッシュ群に、キャッシュＡ６０１のネットワークアドレスとともに、ＵＤＰ　ｐｉｎｇリクエストを送信する。オブジェクトキャッシュ群は、キャッシュＡ６０１にＵＤＰ　ｐｉｎｇヒットメッセージまたはミスメッセージで応答する。

【００３６】本発明の主たる利点はそのスケーラビリティである。ディレクトリがディレクトリリストの形で多数のサーバ間に分配されるため、十分なスケーラビリティをもつ検索機能を実現できる。所与のオブジェクトに対する正しいディレクトリリストは受信キャッシュ中で実行されるハッシュロケータ機能によって見つけだされる。このように、多数のリクエストによる負担を受け、かつ１つのサーバ上に存在する単独の非分散型ディレクトリリストを完全に検索する代わりに、ハッシュロケータ関数と短いディレクトリリストとを参照することによって、ネットワーク上の多数の異なるサーバ上にキャッシュされた多数の異なるオブジェクトを簡単かつ効率的に発見できる。

【００３７】リクエストされたオブジェクトのコピーをもつキャッシュ群のアドレスが判明すると、決定システムが実行されて、リクエストされたオブジェクトのコピーをどのキャッシュから送信するかを決定する。一実施形態では、リクエストされたオブジェクトのキャッシュされたコピーがなければ、受信キャッシュはオブジェクトに対するリクエストをオブジェクトのオリジナル情報源のサーバに送信する。他の実施形態では、受信キャッシュはオブジェクトに対するリクエストを、キャッシングシステムの他の階層レベルの別のキャッシュに送信する。

【００３８】上述したように、本発明の一実施形態では、受信キャッシュからの探索リクエストに応答して、ディレクトリキャッシュから受信キャッシュへオブジェクトキャッシュのアドレス群が送られ、この場合は受信キャッシュがオブジェクトキャッシュ群をポーリングする。他の実施形態では、オブジェクトキャッシュアドレス群は受信キャッシュへは送られず、この場合はディレクトリキャッシュがオブジェクトキャッシュ群をポーリングする。いずれの場合も、オブジェクトキャッシュ群から受信キャッシュへＵＤＰ　ｐｉｎｇヒットまたはミスメッセージが送られ、受信キャッシュはリクエストされたオブジェクトのキャッシュされたコピーの送信をどのオブジェクトキャッシュに依頼するかを決定する。

【００３９】図７は、オブジェクトキャッシュアドレス群が受信キャッシュに送信されない場合の実施形態を示す。キャッシュＡ７０１はクライアントからオブジェク

トYに対するリクエストを受信している。キャッシュＡ７０１はディレクトリ探索関数Ｌ（Ｙ）＝Ｃ（図示せず）を実行し、オブジェクトYのディレクトリリストがキャッシュＣ７０２に記憶されていることを示す。受信キャッシュＡ７０１はディレクトリキャッシュＣ７０２に探索メッセージ（“ＬＯＯＫＵＰ（１）”）を送信する。この例では、キャッシュＣ７０２に記憶されているYのディレクトリリストは、YのコピーがキャッシュＥ７０３とＦ７０４とに記憶されていることを示す。ディレクトリキャッシュＣ７０２は、オブジェクトキャッシュＥ７０３とＦ７０４とにＵＤＰ　ｐｉｎｇリクエストを送信する。図８に示すように、キャッシュＡ８０１はまず、キャッシュＥ８０２からＵＤＰ　ｐｉｎｇヒット応答（“ＵＤＰ＿ｐｉｎｇ＿ｈｉｔ（１）”）を受信し、その後、キャッシュＦ８０３からＵＤＰ　ｐｉｎｇヒット応答（“ＵＤＰ＿ｐｉｎｇ＿ｈｉｔ（２）”）を受信する。図９に示すように、キャッシュＡ９０１が受信した最初のＵＤＰ　ｐｉｎｇヒット応答はキャッシュＥ９０２からであったので、キャッシュＡ９０１はキャッシュＥ９０２にオブジェクトYのコピーのリクエストを送信する。この決定規準の有利な点は、キャッシュＡ９０１がオブジェクトキャッシュ群をポーリングしてからキャッシュＥ９０２からＵＤＰ　ｐｉｎｇヒットメッセージを受信するまでの応答時間のほうが、キャッシュＦ８０３（図８）からＵＤＰ　ｐｉｎｇヒットメッセージを受信するまでの応答時間よりも遅延が少ないことに基づいて、ネットワーク的に受信キャッシュＡ９０１に「一番近い」キャッシュにオブジェクトがリクエストされることである。キャッシュＡ９０１からのリクエストに応答して、キャッシュＥ９０２はキャッシュＡ９０１にオブジェクトYのコピーを送信し、キャッシュＡ９０１はそのコピーを記憶してリクエスト元のクライアントへ転送する。

【００４０】この時点で、オブジェクトYのディレクトリリストはキャッシュＡ９０１がオブジェクトYのコピーを持っていることを反映していないため、不正確となる。本発明の一実施形態では、キャッシュＡ９０１はキャッシュＣ９０３にディレクトリリストを更新するようメッセージを送る。キャッシュＣ９０３はオブジェクトYのディレクトリリストにキャッシュＡ９０１のネットワークアドレスを加える。キャッシュＡ９０１はこうしてYのオブジェクトキャッシュとなる。

【００４１】本発明の他の実施形態では、キャッシュＣ７０２（図７）は、リクエストされたオブジェクトのディレクトリリストにアドレスが掲載されているすべてのキャッシュにはＵＤＰ　ｐｉｎｇリクエストを送信しない。この代わりに、キャッシュＣ７０２はリストから一定数αのアドレス群を選択する。図７に示す例においてオブジェクトYのディレクトリリストはＢ７０５，Ｄ７０６，Ｅ７０３，およびＦ７０４であるとする。キャッシュＣ７０２はαを２に選択し、キャッシュＡ８０１のアドレスとともにＵＤＰ　ｐｉｎｇリクエストを送信するキャッシュとしてディレクトリリストからキャッシュＥ７０３およびＦ７０４を選択する。キャッシュＥ８０２（図８）のＵＤＰ　ｐｉｎｇヒットメッセージが最初にキャッシュＡ８０１によって受信されるので、キャッシュＡ８０１はオブジェクトのリクエスト（“ｒｅｑｕｅｓｔ（１）”（図９））をキャッシュＥに送る。リクエストされたオブジェクトのコピー（“ＣＯＰＹ（２）”）はキャッシュＥ９０２からキャッシュＡ９０１（図９）に送られる。オブジェクトのコピーはキャッシュＡ９０１に記憶され、コピー（“ＣＯＰＹ（３）”）がリクエスト元のクライアント９０４に送信される。キャッシュＡ９０１は、YのディレクトリリストにキャッシュＡ９０１のアドレスを加えて更新するようキャッシュＣ９０３にメッセージを送る。

【００４２】αがディレクトリリストに掲載されたアドレス数より少ないときは常に、キャッシュは、ディレクトリリストのどのキャッシュにＵＤＰ　ｐｉｎｇリクエストを送信するかを決定しなければならない。一実施形態では、有利には、キャッシュはラウンドロビン方式（スケジューリング）を用いて、リストに掲載されたキャッシュ間に負担を分散できる。他の実施形態では、キャッシュはリクエストを送るキャッシュ数αをランダムに選択する。

【００４３】本発明の一実施形態では、ディレクトリキャッシュはＵＤＰ　ｐｉｎｇリクエストを送るキャッシュをディレクトリリストから１つ、つまりα＝１に選択する。例えば、キャッシュＣ１００１（図１０）がキャッシュＥ１００２を選択するとする。キャッシュＣ１００１はキャッシュＡ１００３のアドレスとともにＵＤＰ　ｐｉｎｇリクエストメッセージをキャッシュＥ１００２に送り、キャッシュＥ１００２は直接キャッシュＡ１００３に応答（ＵＤＰ　ｐｉｎｇヒットまたはミス）を送信する。もし応答がＵＤＰ　ｐｉｎｇヒットならば、キャッシュＡ１００３はオブジェクトのリクエストをキャッシュＥ１００２に送る。もし応答がＵＤＰ　ｐｉｎｇミスならば、ディレクトリリストは正しくないから、キャッシュＡ１１０１（図１１）はキャッシュＣ１１０２にディレクトリ削除リクエストを送り、キャッシュＣ１１０２はリクエストされたオブジェクトのディレクトリリストからキャッシュＥ１１０３のアドレスを削除する。一方、キャッシュＥがキャッシュＡからのオブジェクトリクエストに応答しない場合は、キャッシュＥ、またはキャッシュＥへの接続が作動不能かもしれないので、リスト上の別のキャッシュを選択してＵＤＰ　ｐｉｎｇリクエストを送信しなければならない。

【００４４】α＝１の場合、コピーの検索に必要なメッセージの数は、リクエストされたオブジェクトのコピーをキャッシュＣ１００１からキャッシュＡ１００３に運

ぶメッセージを除き、有利なことに４つに減らされる。これは、同じ目的のために２Ｎ＋１個のメッセージが必要な、図２～図４に示す従来の決定関数よりはるかに効率的である。本発明に従ってα＝１について決定関数を実行することにより、接続または装置の故障の確率が非常に低く、かつキャッシュＣ１００１が全キャッシュ間に負担を分散する効率的な方法を実行する信頼性のあるネットワークでは、良好なキャッシュ性能が得られる。決定システムが実行されるとディレクトリリストが更新される方式は、信頼性の高いネットワーク（信頼性の高い装置および接続をもつネットワーク等）について有効に動作するため、「楽観的アプローチ」と呼ばれる方式を示す。

【００４５】ネットワークの中には信頼性が完全ではないものがある。かかるネットワークの一例はインターネットであり、所与の装置または接続が動作しているかどうか、またはある任意の時間に利用不能かどうかの予測が困難または不可能である。さらに、ディレクトリキャッシュと受信キャッシュとがデータオブジェクトに対するリクエストをどこに送るかについてそれぞれ独自に決定を行うため、負担をキャッシュ間に効率的に分散するのが困難である。つまり、負担を均等に分配しようとしているキャッシュが、負担が他のキャッシュ群に独自にどのように分配されているかを知り得ない場合があり、この結果、データオブジェクトのリクエストを１つ以上のキャッシュに予想外に集中させてしまう。α＝１を選択しても、選択されたキャッシュまたは該キャッシュへの接続が一時的に作動不能になっていて、リクエストされたオブジェクトを該キャッシュが送信できない場合があるために効率的でない。従って、有利には「悲観的アプローチ」と呼ばれる方式が実行され、この方式では、決定関数の実行中はディレクトリリストは更新されず、ただしキャッシュが新しいオブジェクトを受けとるとディレクトリリストに適切なアドレスが加えられ、キャッシュがオブジェクトを排出（削除）するとそのアドレスがディレクトリリストから削除される。

【００４６】α＝１で、かつオブジェクトリクエストに対する応答が否定的な場合（ＵＤＰｐｉｎｇミス応答が受信されるか、または応答がない場合等）は、別のキャッシュが選択され、まったく新しいＵＤＰ　ｐｉｎｇメッセージの組がやりとりされて、リクエストされたオブジェクトを可能ならば他のキャッシュに送信させる。α＝１を選択すると、ディレクトリリストが不正確で、選択された１つのキャッシュが実際はリクエストされたオブジェクトのコピーを持っていないという危険もある。α＞１に選択することで、オブジェクトのコピー入手過程に冗長性をもたせ、リクエストされたオブジェクトを１つ以上の可能性のある場所で探索することによって、信頼性のないネットワーク中での上記の問題を軽減することができる。またこの冗長性により、悲観的アプローチでは訂正されないいくつかの不正確なディレクトリエントリをもつ影響を緩和できる。

【００４７】本発明の一実施形態では、αに定数Ｋを選択する。この場合、ディレクトリリスト上のキャッシュ数ＮがＫ以下ならば、ＵＤＰ　ｐｉｎｇリクエストはリクエストされたオブジェクトのコピーを記憶しているＮ個のキャッシュ、および該コピーを記憶していないＫ－Ｎ個のキャッシュへ送信され、本発明の効率は従来のキャッシングシステムのものに近づく。ディレクトリリスト上のキャッシュ数ＮがＫ以上ならば、キャッシュはディレクトリリストからＫ個のキャッシュを選び出し、各キャッシュにＵＤＰ　ｐｉｎｇリクエストを送る。この場合、作成されるメッセージ数は、リクエストされたオブジェクトのコピーを運ぶメッセージを除いて２Ｋ＋３以下となる。これは上記でα＝１について議論したのと同じ理由で信頼性のあるネットワーク中で有効となりうる。しかし信頼性のないネットワークでは、リクエストされたオブジェクトのコピーをもっていると称するディレクトリリスト上の全キャッシュ群にＵＤＰ　ｐｉｎｇリクエストを送信することによって得られる冗長性を完全には利用できない。

【００４８】本発明に従えば、α値は、有利には本発明が実現されるネットワークの信頼性に従ってキャッシングシステムの性能を最大にするべく選択される。信頼性の高いネットワークでは、αは１に近く選択される。信頼性の低いネットワークでは、αは１より大きく選択され、ネットワークの信頼性が低いほどより大きな値が選択される。ただし信頼性のないネットワークでのαの値は、多くてもディレクトリリストのアドレス数と同じにしなければならない。信頼性のないネットワーク上で実現される本発明の一実施形態では、各ディレクトリリストのサイズはＫ個のアドレスに固定され、αはＫに設定される。本発明の他の実施形態では、αは全ディレクトリリスト群の平均アドレス数に設定される。さらに他の実施形態では、αは全ディレクトリリスト群のメジアン数に設定される。楽観的アプローチでは、あるキャッシュがオリジナルコピーを受けとった場合、すなわちオブジェクトのオリジナル情報源にオブジェクトのリクエストを送信した場合にのみ、キャッシュはオブジェクトのディレクトリリストを記憶しているキャッシュにディレクトリ追加リクエストを送信する。ディレクトリ追加リクエストを受信すると、ディレクトリキャッシュは、所与のオブジェクトのディレクトリリストに送信元のキャッシュのアドレスを追加する。また楽観的アプローチでは、所与のオブジェクトのコピーをリクエストするキャッシュがオブジェクトキャッシュからリクエストされたオブジェクトのコピーが受信できなかった場合は、リクエスト元のキャッシュはディレクトリキャッシュにディレクトリ削除リクエストを送信し、該オブジェクトのディレクトリリストからそのオブジェクトキャッシュのア

ドレスを削除する。また、オブジェクトキャッシュがオブジェクトを排出するたびに、オブジェクトキャッシュから適当なディレクトリキャッシュへディレクトリ削除リクエストが送信され、これによりオブジェクトキャッシュのアドレスがディレクトリリストから削除される。

【００４９】悲観的アプローチでは、キャッシュがどの情報源からでも所与のオブジェクトの新しいコピーを受信するたびに、キャッシュはディレクトリキャッシュにディレクトリ追加リクエストを送信する。リクエスト元のキャッシュがリクエストされたコピーが入手できなくても、ディレクトリ削除リクエストは送信されない。ディレクトリ削除リクエストは、オブジェクトを記憶しているキャッシュがそのオブジェクトを排出するときにのみ適当なディレクトリキャッシュに送信される。このディレクトリ削除リクエストはオブジェクト排出元のキャッシュから送信される。削除リクエストを受信すると、オブジェクト排出元キャッシュのアドレスがディレクトリリストから削除される。

【００５０】図１２に示すように、データオブジェクトは、古くなると、記憶されているキャッシュから排出されなければならない。これは本発明の一形態に従って、生存時間（ＴＴＬ）パラメータを記憶されている各オブジェクトに関連づけることによって行われる。ＴＴＬパラメータはキャッシュされたオブジェクトの最大寿命を特定する。一実施形態では、ＴＴＬパラメータは、オブジェクトのコピーが受信されキャッシュに記憶された時からある一定の時間量だけずれた日付スタンプである。ＴＴＬが期限切れすると、オブジェクトは排出される。ＴＴＬパラメータは従来のシステム中で実現されており、古くなったコピーをキャッシュ中に残留させてしまうという欠点がある。従来のシステムでは、この問題は所与のオブジェクトが古くなる（オリジナルが変更される等）と、該オブジェクトのすべてのキャッシュに排出メッセージを配信することによって対処される。しかし、排出メッセージの配信は、ネットワークに不利に負担をかける大量のトラフィックを生じてしまう。

【００５１】本発明は、古くなったキャッシュされたコピーを効率的に排出して、費用のかかる配信メッセージの使用を避けることによって、この問題を解決する。本発明に従えば、オリジナルオブジェクトが変更されるたびにオブジェクトのオリジナル情報源からディレクトリキャッシュに排出リクエストを送信することによって、オブジェクトをそのＴＴＬが期限切れする前に排出できる。その後、ディレクトリキャッシュは、オブジェクトのディレクトリリスト上の全オブジェクトキャッシュに排出メッセージを送信する。これを受けて、オブジェクトキャッシュはオブジェクトを排出し、ディレクトリリスト自体がディレクトリキャッシュから削除される。

【００５２】この様子を図１２に示す。キャッシュＡ１２０１は、オリジナル情報源サーバ１２０２からオブジェクトＹの全コピーの排出リクエストを受信する。キャッシュＡ１２０１はディレクトリロケータ関数Ｌ（Ｙ）＝Ｃ１２０３を実行し、オブジェクトＹのディレクトリリストがディレクトリキャッシュＣ１２０４に記憶されていることを知る。キャッシュＡ１２０１はキャッシュＣ１２０４にディレクトリ排出リクエストを送信する（ｄｉｒ＿ｅｊｅｃｔ（１））。キャッシュＣ１２０４上のディレクトリリストＹは、ＹのキャッシュされたコピーがキャッシュＥ１２０５およびＦ１２０６に記憶されていることを示す。ディレクトリ排出リクエストを受信すると（ｄｉｒ＿ｅｊｅｃｔ（２））、キャッシュＣ１２０４はキャッシュＥ１２０５およびＦ１２０６に排出リクエスト（ｅｊｅｃｔ（３））を送り、これらのキャッシュはＹのコピーを排出する。その後、キャッシュＣ１２０４はそのＹのディレクトリリストを削除する。他の実施形態では、キャッシュＡは、オブジェクトのオリジナル情報源サーバからオブジェクトのコピーを受信するたびに、ディレクトリサーバにディレクトリ排出リクエストを送信する。この実施形態はオリジナル情報源サーバからの排出メッセージを必要としない。本発明は、従来のいくつかのキャッシングシステムで実行された全ディレクトリキャッシュへの排出メッセージの配信を不要にできる。これにより古くなったオブジェクトのコピーの排出のために生じるネットワークトラフィック量を大幅に削減でき、キャッシングシステムの正確さをより効率的に維持できる。

【００５３】本発明に従って実現される分散型排出方式は、当該技術分野で公知である全キャッシュへの排出メッセージの配信を不要にする。分散型排出方式は、古くなったオブジェクトの排出に必要なネットワークトラフィックだけしか生じないため、配信方式よりも効率的かつ経済的である。さらに、分散型排出は大規模なシステムにも小規模なシステムと同様に有効に動作するので、ネットワークのサイズが大きくなるとそれに応じて負担となる大量のトラフィックを生じる配信方式よりもスケーラブルである。

【００５４】本発明は、従来のキャッシングシステムと比べて効率が高く、かつスケーラブルなキャッシングシステムを提供する。このシステムは、ディレクトリリストの形の分散型ディレクトリを提供し、各ディレクトリリストは１つのデータオブジェクトに関連づけられる。ディレクトリリストは、オブジェクトのコピーが記憶されているオブジェクトキャッシュ群のリストである。オブジェクトのディレクトリリストはネットワークの多数のサーバ間に分散される。所与のオブジェクトのディレクトリリストの位置は、オブジェクトのネットワークアドレスをディレクトリリストが記憶されているディレクトリキャッシュのネットワークアドレスを指すポインタに相関させるハッシュロケータ関数によって、簡単かつ経済的に求められる。ディレクトリリストの多数のオブ

ジェクトキャッシュ群がポーリングされ、その中からキャッシュされたコピーを取り出す１つのキャッシュが選択される。オブジェクトキャッシュの数は、信頼性の異なる様々なネットワーク環境中でキャッシングシステムの性能を最大にするように、本発明に従って有利に選択できる。ディレクトリリストは、ネットワークの状態に従って分散式で更新される。古くなったオブジェクトおよびディレクトリは、効率的な分散方式でキャッシングシステムから排出される。本発明は、従来のキャッシングシステムに比べて高い信頼性と低い費用で実現することができる。さらに、本発明はボトルネックや単一故障点の影響を受けにくいので、従来のキャシングシステムに関連した多くの問題を解決でき、ネットワークにかけるキャッシングシステムのネットワークトラフィックの負担を従来のシステムより大幅に削減できる。

【００５５】以下に、本発明の様々な形態の仮想的なコードの実施例を示す。所与のオブジェクトのコピーの入手を望むクライアントは、あるキャッシュに入手リクエストメッセージを送信する。図５に示すように、オブジェクトＹに対する入手リクエストはクライアント５０２からキャッシュＡ５０１に送られる。入手リクエストの仮想的なコードの実施例は図１３のようになる。

【００５６】上記の入手リクエストの実施例はディレクトリ探索リクエストを実行する。探索リクエストの仮想的なコードは図１４のようになる。

【００５７】ディレクトリリストからα個のアドレスを選択するディレクトリ探索ルーチンの仮想的なコードの実施例は図１５のようになる。

【００５８】ＵＤＰ　ｐｉｎｇリクエスト／ヒット／ミス・ルーチンの仮想的なコードの実施例は図１６のようになる。

【００５９】ディレクトリ追加ルーチンの仮想的なコードの実施例は図１７のようになる。

【００６０】ディレクトリ削除ルーチンの仮想的なコードの実施例は図１８のようになる。

【００６１】排出ルーチンの仮想的なコードの実施例は図１９のようになる。

【００６２】オリジナル情報源からオブジェクトのコピーを入手するリフレッシュルーチンの仮想的なコードの実施例は図２０のようになる。

【００６３】応答が目的地から受信されたかどうかをテストし、適当な維持方式を実行するルーチンの仮想的なコードの実施例は図２１のようになる。

【００６４】本発明に従う分散型排出方式の仮想的なコードの実施例は図２２のようになる。

【００６５】排出ルーチンの仮想的なコードの実施例は図２３のようになる。

【００６６】本発明は、有利には、従来のキャッシングシステムより効率的な、オブジェクトのキャッシングシステムおよび方法をネットワーク上で提供する。本発明に従う分散型方法は、有利には、キャッシュされたオブジェクトを記憶、発見、入手、および排出を行うスケーラブルな方法を提供し、ネットワーク上のデータオブジェクト数が増大しても、ネットワークに課すキャッシングシステムトラフィックの負担を低く抑えつつ性能を向上させることができる、改良されたネットワークを提供する。本発明はまた、データオブジェクトが頻繁に変化するネットワークにおいて、従来のキャッシングシステムよりも優れた性能を発揮する。これは、有利には従来のキャッシングシステムにおいて不都合にネットワークに負担をかけていたロッキング、承認、および配信メッセージを必要とすることなく達成できる。

【図面の簡単な説明】

【図１】　先行技術のキャッシングのシステムの実施形態を示す図である。

【図２】　先行技術のキャッシングのシステムにおけるＵＤＰ　ｐｉｎｇリクエストの伝達過程を示す図である。

【図３】　先行技術のキャッシングのシステムにおけるＵＤＰ　ｐｉｎｇヒットメッセージおよびＵＤＰ　ｐｉｎｇミスメッセージの伝達過程を示す図である。

【図４】　先行技術のキャッシングのシステムにおいて、オブジェクトのコピーに対するリクエストを伝達し、リクエストされたオブジェクトのコピーをキャッシュおよびクライアントに伝達する過程を示す図である。

【図５】　本発明に従うキャッシングのシステムの一実施形態において、クライアントがオブジェクトに対するリクエストを受信キャッシュに送信し、受信キャッシュがリクエストされたオブジェクトに対してディレクトリロケータ関数を実行する様子を示す図である。

【図６】　本発明に従うキャッシングのシステムの一実施形態において、クライアントからのオブジェクトに対するリクエストに応答して、受信キャッシュがディレクトリキャッシュに探索リクエストを送信する様子を示す図である。

【図７】　本発明に従うキャッシングのシステムの一実施形態において、ディレクトリキャッシュが、リクエストされたオブジェクトのディレクトリリスト上のキャッシュ群にＵＤＰ　ｐｉｎｇリクエストを送信する様子を示す図である。

【図８】　本発明に従うキャッシングのシステムの一実施形態において、リクエストされたオブジェクトのディレクトリリスト上のキャッシュ群が、ディレクトリキャッシュに対してＵＤＰ　ｐｉｎｇヒットメッセージで応答する様子を示す図である。

【図９】　本発明に従うキャッシングのシステムの一実施形態において、受信キャッシュがオブジェクトリクエストメッセージをディレクトリリスト上のキャッシュに送信し、このキャッシュがリクエストされたオブジェクトのコピーを受信キャッシュに送信し、受信キャッシュ

がオブジェクトのコピーをクライアントに送信する様子を示す図である。

【図１０】　本発明に従うキャッシングのシステムの一実施形態において、オブジェクトキャッシュ群の１つのキャッシュにだけ、オブジェクトリクエストメッセージが送信される様子を示す図である。

【図１１】　本発明に従うキャッシングのシステムの一実施形態において、キャッシュがディレクトリキャッシュに削除リクエストメッセージを送信する様子を示す図である。

【図１２】　本発明に従うキャッシングのシステムの一実施形態において、サーバが受信キャッシュに所与のオブジェクトに対するディレクトリ排出メッセージを送信し、受信キャッシュがこのオブジェクトに対してディレクトリロケータ関数を実行し、かつディレクトリキャッシュにディレクトリ排出メッセージを送信し、ディレクトリキャッシュがオブジェクトに対してディレクトリリスト上のキャッシュに排出メッセージを送信する様子を示す図である。

【図１３】　入手リクエストの仮想コードを示す図である。

【図１４】　探索リクエストの仮想コードを示す図である。

【図１５】　ディレクトリ探索ルーチンの仮想コードを示す図である。

【図１６】　ｐｉｎｇリクエストの仮想コードを示す図である。

【図１７】　ディレクトリ追加ルーチンの仮想コードを示す図である。

【図１８】　ディレクトリ削除ルーチンの仮想コードを示す図である。

【図１９】　排出ルーチンの仮想コードを示す図である。

【図２０】　リフレッシュルーチンの仮想コードを示す図である。

【図２１】　レスポンス待ちルーチンの仮想コードを示す図である。

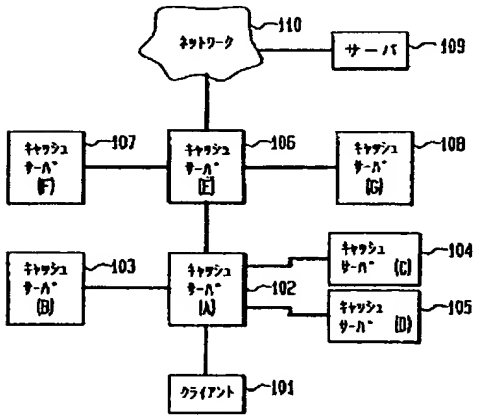
【図２２】　ディレクトリ排出ルーチンの仮想コードを示す図である。

【図２３】　排出ルーチンの仮想コードを示す図である。

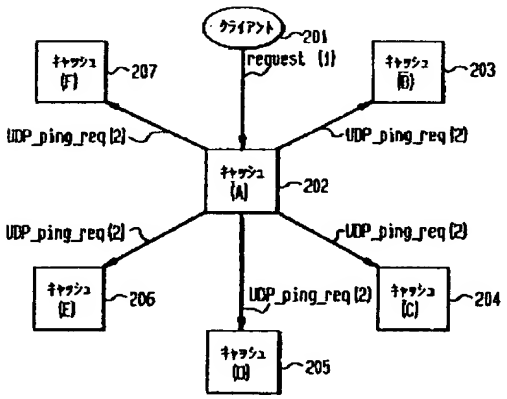
【符号の説明】

５０１　受信キャッシュ、５０２　クライアント、５０３　ディレクトリキャッシュ、５０４，５０５，５０６，５０７　オブジェクトキャッシュ、５０８ハッシュロケータ関数。

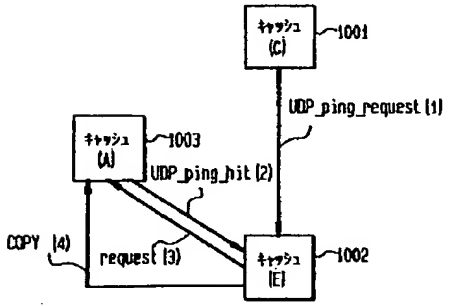
【図１】



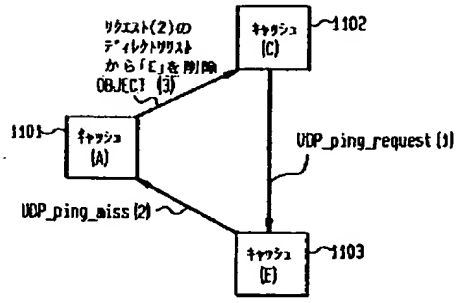
【図２】



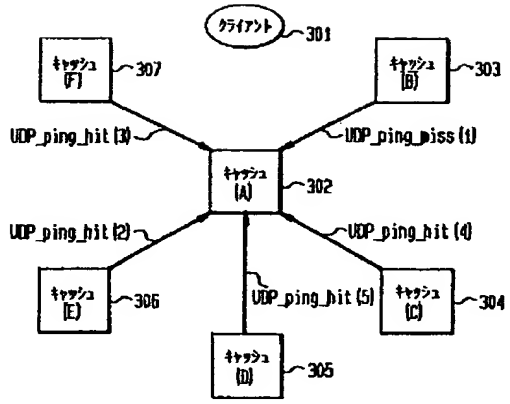
【図１０】



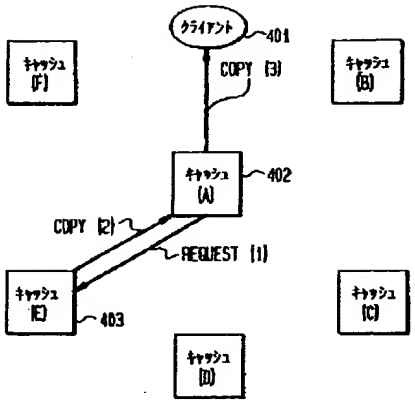
【図１１】



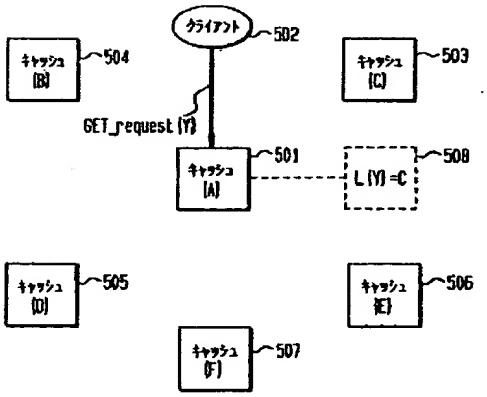
【図３】



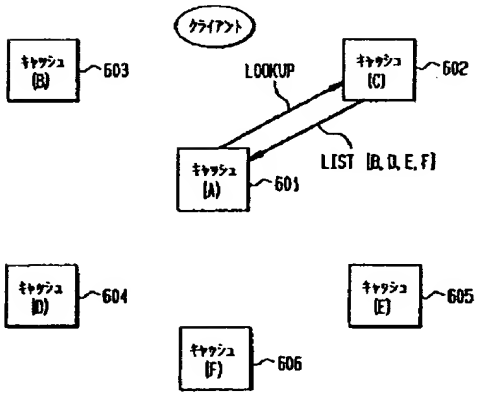
【図４】



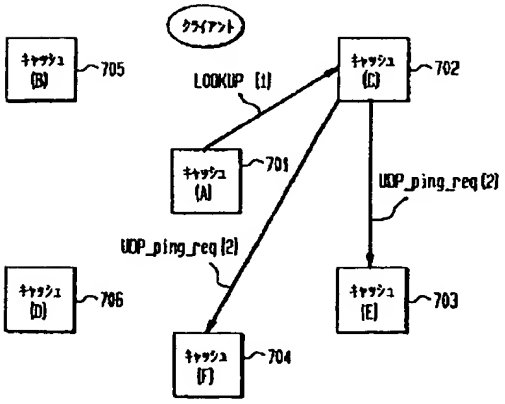
【図５】



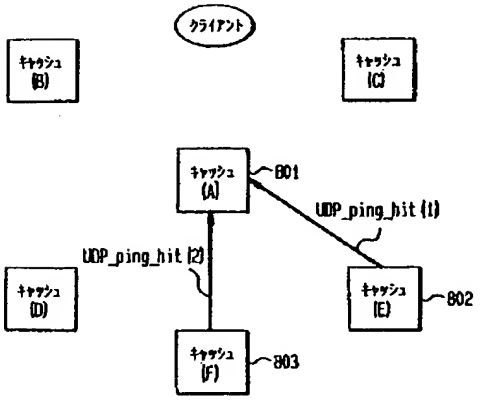
【図６】



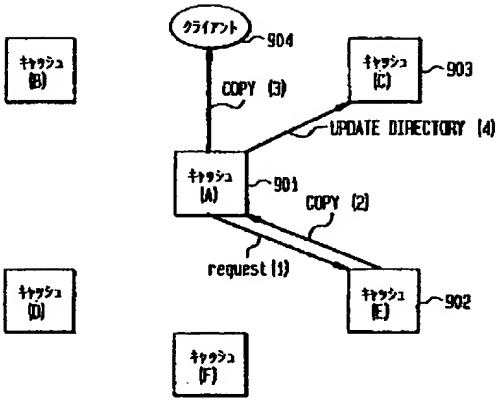
【図７】



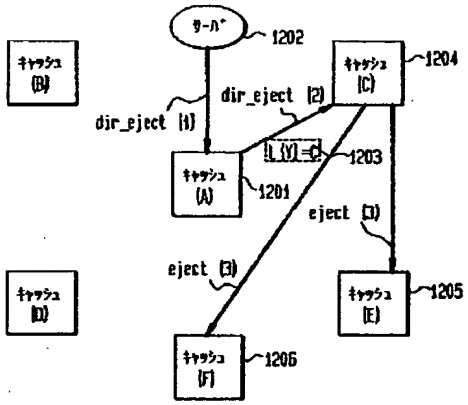
【図８】



【図９】



【図１２】



【図１４】

```
dir_lookup (input:url_of_object, client_IPaddr;output:list_of_neighbors )
                    /*client は探索リクエストの送信元を指す */
{
find_URL(url_of_object);
 /*リクエストされたオブジェクトのネットワークアドレスに関連
   したディレクトリリストを見つける */

if URL_of_object is found
then return(list_of_neighbors);
 /*もしリクエストされたオブジェクトのディレクトリリストが見
   つかれば、ディレクトリリスト上のアドレスリストを返送 */

else return(null_list);

if optimistic approach
  find_CLIENT(client_IPaddr);
 /*楽観的アプローチにおいて、もし顧客キャッシュアドレスがま
   だディレクトリリスト上になければ、顧客キャッシュのアドレ
   スを加えてディレクトリリストが更新される */

   if client_IPaddr is not found
     add_neighbor_to_list(url_of_object, client_IPaddr);
}
```

【図２３】

```
eject (input:url_of_object)
{
find object url_of_object in local cache;
if object is found
then remove object url_of_object from the local cache;
}
```

【図１３】

```
GET_request (input: url_of_object )
    /*   url_of_object はリクエストされたオブジェクトのネットワーク
      アドレス */
{
...
dest_IPaddr:=HASH_TABLE_NEIGHBORS[hash_1(url_of_object)];
    /* この行は本実施例でオブジェクトのネットワークアドレスに
       対して実行されるハッシュ関数であるディレクトリロケータ
       関数を示す */
if dest_IPaddr is equal to my_IPaddr
                /* ロケータ関数がディレクトリキャッシュ自身を指し
                   ているか？ */
then lookup(url_of_object, my_IPaddr);
                /* ロケータ関数がディレクトリキャッシュを指して
                   いれば、リクエスト元の顧客にコピーを送信する
                   探索ルーチンが実行される */
else send(dest_IPaddr, lookup(url_of_object, my_IPaddr);
  /* この行は探索関数で示されたディレクトリリストが位置する、
    受信キャッシュ以外のキャッシュ（「ディレクトリキャッシュ」）
    に探索リクエストを送信する */
loop until(response is equal to UDP_ping_hit
          or response is equal to NO_neighbors
          or waiting_time expired
          or number _of_responses is equal to alpha)
     wait(response, neighbor_IPaddr);
  /* このループは、最初の UDP ping 当たりメッセージが受信される
    まで、または受信キャッシュが直接通信可能なキャッシュに
    リクエストされたオブジェクトのコピーが記憶されていないと
    いうメッセージになるまで、または所与の待機時間が応答なし
    で時間切れするまで、または応答数が所定数αになるまで続行
    する */
if response is UDP_ping_hit
then send(neighbor_IPaddr, request(url_of_object, my_IPaddr));
/*UDP ping ヒットを受信すると、リクエストされたオブジェクトを
  ネットワークアドレス my_IPaddr の受信キャッシュに送信するよう
  に UDP ping ヒットの送信側にメッセージが送信される */
else send(home_url_IPaddr, Get_request(url_of_object, my_IPaddr);
  /* もし UDP ping リクエストが受信されなければ、 home_url_IPaddr の
    リクエストされたオブジェクトの元リースが、リクエストされ
    たオブジェクトのコピーを受信キャッシュに送信するよう依頼
    される */
...
)
```

【図１５】

```
lookup (input:url_of_object, requester_IPaddr )
{
dir_lookup(url_of_object, requester_IPaddr, list_of_neighbors);

if list_of_neighbors is null_list
then send(requester_IPaddr, (NO_neighbors, my_IPaddr));
else
     make alpha_list by choosing alpha neighbors from list_of_neighbors;
     send(alpha_list, UDP_ping_req (requester_IPaddr));
}
```

【図１６】

```
UDP_ping_req (input:url_of_object, client_IPaddr)
                  /*client は UDP ping request の送信元を指す */
{
find object url_of_object in local cache;

if object is found
then send(client_IPaddr, UDP_ping_hit (my_IPaddr));
else send(client_IPaddr, UDP_ping_miss (my_IPaddr));
}
```

【図１７】

```
dir_add (input:url_of_object, client_IPaddr )
   /*client はオブジェクトのコピーが加えられたキャッシュを指す */
{
find_URL (url_of_object);
if url_of_object is not found
   add_new_list (url_of_object);   /* 新しいディレクトリを開始 */
find_CLIENT (client_IPaddr);
   /* ディレクトリリスト上の顧客ネットワークアドレスを探す */
if client_IPaddr is not found
   add_neighbor_to_list (url_of_object, client_IPaddr);
   /* もし見つからなければこれをディレクトリリストに追加 */
}
```

【図１８】

```
dir_del (input:url_of_object, client_IPaddr )
  /*client はオブジェクトのコピーが削除されているキャッシュを
  指す */
{
find_URL (url_of_object);
  /*ディレクトリリスト上のオブジェクトのアドレスを探す */
if url_of_object is found
  /*アドレスが見つかれば、そのアドレスをディレクトリリストか
    ら削除 */
    find_CLIENT (client_IPaddr);
    if client_IPaddr is found
       delete_neighbor_from_list (url_of_object, client_IPaddr);)
}
```

【図１９】

```
eject_local_copy (input:url_of_object )
{
...
dest_IPaddr:=HASH_TABLE_NEIGHBORS [hash_1(url_of_object)];
 /*この行はオブジェクトのディレクトリリストが記憶されている
   キャッシュの場所を突き止めるためのハッシュ関数としてディ
   レクトリロケータ関数を実行する */

if dest_IPaddr is equal to my_IPaddr
 /*ディレクトリキャッシュがローカルキャッシュならば、ローカ
   ルキャッシュのディレクトリリストを削除 */
then dir_del(url_of_object, my_IPaddr);
else send(dest_IPaddr, dir_del(url_of_object, my_IPaddr));
 /*これ以外の場合はディレクトリ削除リクエストをディレクトリ
   キャッシュに送信 */
}
```

【図２０】

```
refresh (input:url_of_object )
{
...
dest_IPaddr:=HASH_TABLE_NEIGHBORS [hash_1(url_of_object)];
            /*ディレクトリロケータ関数 */
if optimistic approach
 /*楽観的アプローチでは、リフレッシュルーチンが実行されると
   ディレクトリリストを更新する */

  if dest_IPaddr is equal to my_IPaddr
  then dir_add(url_of_object, my_IPaddr);
  else send(dest_IPaddr, dir_add(url_of_object, my_IPaddr));
}
```

【図２１】

```
wait_response(input:status)
{
--
dest_IPaddr:=HASH_TABLE_NEIGHBORS[hash_1(url_of_object)];
        /* この行はディレクトリロケータ関数である */
if status is OK and pessimistic approach
 /* 悲観的アプローチにおいてコピーが受信されていれば、受信キ
   ャッシュのアドレスを加えてディレクトリリストが更新される */

   if dest_IPaddr is equal to my_IPaddr
   then dir_add(url_of_object, my_IPaddr);
   else send(dest_IPaddr, dir_add(url_of_object, my_IPaddr));
if status is not OK and optimistic approach
 /* 楽観的アプローチにおいてコピーが受信されていなければ、
    受信キャッシュのアドレスを削除してディレクトリリストが
    更新される */
   if dest_IPaddr is equal to my_IPaddr
   then dir_del(url_of_object, my_IPaddr);
   else send(dest_IPaddr, dir_del(url_of_object, my_IPaddr));
}
```

【図２２】

```
dir_eject(input:url_of_object )
{
dest_IPaddr:=HASH_TABLE_NEIGHBORS[hash_1(url_of_object)];
                    /* この行はディレクトリ探索関数である */

if dest_IPaddr is not equal to my_IPaddr
                    /* まずローカルキャッシュをチェックする */
then send(dest_IPaddr, dir_eject(url_of_object));
else
  find_URL(url_of_object);          /* ディレクトリリストを見つける */
  if url_of_object is found
  then
     send(list_of_neighbors, eject(url_of_object));
     clear the list_of_neighbors;   /* ディレクトリリストをクリアする */

eject(url_of_object);               /* ローカルコピーを排出する */
}
```

フロントページの続き

(72)発明者　パーサ　ピー　ドゥッタ
アメリカ合衆国　カリフォルニア州　サン
ノゼ　マリブ　ドライブ　1164

(72)発明者　トーマス　ビー　ロンドン
アメリカ合衆国　カリフォルニア州　マウ
ンテン　ビュー　ラモス　コート　2739

(72)発明者　シニサ　サーブルジック
クロアチア　ヴァリカ　ゴリカ　10410
ジョシパ　ジェラシオス　91　ヴェーネ
ユー１

(72)発明者　ダリボー　エフ　ヴルサロヴィック
アメリカ合衆国　カリフォルニア州　サニ
ーヴェイル　カムサック　コート　932